REMARKS

Introduction

The Office action dated August 3, 2005, and the references cited therein have been carefully considered. In response, the applicant offers the following remarks.

Declaration under 37 C.F.R. § 1.131

The Office action stated that the original declaration dated July 19, 2004 was ineffective to overcome Spasojevic, U.S. Patent 6,269,410 (hereinafter "Spasojevic"). In response, the applicant submits a new declaration under 37 C.F.R. § 1.131 to overcome this objection. The new declaration fully considers the relationship between the 1998 article entitled "Network Performance Monitoring in Windows NT" and the current application. The handwritten notations and markings in the copy of this article submitted with the new declaration are irrelevant for purposes of this action. Because the new declaration effectively swears behind Spasojevic, that prior reference is not considered in the § 102(e) discussion set forth below.

Office Action Objection

Claim 3 was objected to due to a typographical error. In response, claim 3 is currently amended to correct the error.

Claim Amendments

Claim 10 is amended to correct a typographical error.

Office Action Rejections

Claims 1-22 and 24 are pending and were rejected under 35 U.S.C. 102(e) and 103(a) in view of Agarwal, U.S. Patent 5,958,010 (hereinafter "Agarwal") and Agarwal in view of Khanna, U.S. Patent 5,978,849 (hereinafter "Khanna"). Additionally, claims 4-6, 10, 15, 17, and 18 were rejected under 35 U.S.C. 102(e) as anticipated by Spasojevic based on the Office action's review of the 37 C.F.R. § 1.131 declaration.

Summary of the Prior Art

Agarwal discloses a system for automated monitoring and management of distributed applications, client/server databases, networks and systems across a heterogeneous environment. Specifically, the invention employs distributed,

automated intelligent monitoring agents with embedded sensing technology to continuously monitor the network environment in real time. In operation, the monitoring agent is coupled to the communications stack for monitoring the data that is being passed between the client and the network. The data collected may then be used for, among other things, network resource allocation.

Khanna discloses a system which facilitates establishing a TCP connection between a client and a server utilizing a control block associated with a previous connection in a "TIME-WAIT" state between the client and the server.

Spasojevic discloses recording activity of a data storage system by collecting system traces generated during the I/O activity, grouping records in the system traces according to stores, identifying I/O activity in streams corresponding to the stores, and processing the groups of records to characterize I/O activity for the streams. The stores represent units of storage such as logical drives, single data storage devices and groups of data storage devices in the data storage system. This data may then be used to allocate network resources.

Arguments for Patentability under § 102(e)

An issue of patentability raised by the Office action and which the applicant must overcome is whether Agarwal anticipates the claimed invention. The applicants submit that claims 1-6, 10-18, 21, 22, and 24 meet the requirements of patentability under 35 U.S.C. § 102(e) and are therefore allowable.

In relevant part, § 102(e) states that, "a person shall be entitled to a patent unless the invention was described in ... a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent...." 35 U.S.C. § 102(e). In practice, a claim is anticipated by a reference under § 102(e) only if "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631 (Fed. Cir. 1987), MPEP § 2131 (emphasis added).

The Office action rejects claims 1-6, 10-18, 21, 22, and 24 as anticipated by Agarwal's systems and methods. All pending claims require detection or receipt of data at the transport level as explained in claim 1 of the present application:

> A method of tracing data traffic on a network, the method comprising: at the transport layer of a protocol stack residing on a first device in the network...

The Office action refers to Col. 9, lines 1-25 as disclosing this element of independent claims 1, 10, and 12. However, Agarwal does not teach event tracing at the transport layer of a protocol stack. Rather, Agarwal discloses the use of agent modules which attach <u>on top of</u> the TCP device to monitor traffic in and out of managed nodes. Furthermore, the Agarwal monitoring agents are only present within the TCP data stream rather than at the transport control level as claimed in claims 1, 10, and 12. In relevant part, Agarwal states:

> The TapFilter device 162 <u>attaches on top of the TCP device 167 of the TCP/IP driver 165 to monitor all TCP traffic in and out of the managed node.</u> The TapFilter device 162 uses a feature of the Windows NT driver layering architecture to attach itself to the TCP device 167. In particular, the TapFilter device 162 uses an operating system call (i.e., IoAttachDevice)<u> to insert itself into the data stream for the TCP device. Thus, any data to or from the TCP device passes transparently through the TapFilter device</u> 162. The System Services 176, the TAP driver 163, and the TCP/IP driver may execute in kernel mode, as shown in FIG. 5.

Agarwal does not disclose the independent claim 1, 10, and 12 element of monitoring at the transport level as disclosed by the present application. All monitoring activity taught by Agarwal involves autonomous agent structures which attach "on top of" TCP devices and log network activity as data passes "in and out of the managed node." Col. 8, lines 12-13 and Col. 9, lines 15-25.

Furthermore, the Office action indicates Agarwal discloses the claim 10 element of a facility for tracing data traffic on a network at the transport layer, the facility comprising: an identifying means for identifying a process causing a transport-layer request to transmit via the network. Agarwal does not describe "tracing data traffic on a network at the transport layer" or "identifying processes causing transport-layer requests to transmit" as the present application claims, but rather, discloses autonomous agents which can be multi-threaded processes monitoring component and transaction traffic taking place outside of the transport layer. These agents, in turn, report to a central console. Col. 6, lines 10-30 state:

> Each of the <u>MUM agents 30-40 can operate autonomously to monitor multiple components</u> including local processor performance, local processor resources, local processor configuration, the operation of the distributed application, the operation of the network, the operation of the various network devices including disks, file systems devices and tape, and other such information. Accordingly, it is a realization of the invention that diagnostic analysis can employ more than a measure of server CPU performance. Each of the <u>monitoring agents 30-40 can be a single, multi-threaded process</u> that can contain information to avoid constant polling by the enterprise monitor console 42. Each of the agents can continuously monitor in real time business <u>transactions, databases, systems and networks detecting and correlating events</u>, initiating corrective actions, and providing event notifications. The <u>MUM agents</u> are capable of understanding business transactions and can collect details of events, resource usage and response times and can pass this information to the MUM console 42 and store it in the MUM database 44.

Because Agarwal's "MUM agents" monitor traffic autonomously from outside the transport layer, Agarwal does not teach the claim 10 element of a facility for tracing data traffic on a network at the transport layer, the facility comprising: an identifying means for identifying a process causing a transport-layer request to transmit via the network.

Likewise, Agarwal does not teach tracing data traffic as claimed in claim 10. Instead, Agarwal describes collecting data regarding the operation of the local system itself. Furthermore, Agarwal's "system monitor" is part of the autonomous MUM agent operation which, as previously explained, exists <u>outside</u> of the transport layer, not <u>at</u> the transport layer as claimed in claim 10. At Col. 7, lines 34-45, Agarwal states:

> In particular, <u>the system monitor 70 can collect information about the operation of the local system</u>. To this end, the <u>system monitor</u> 70 can include code modules for collecting information regarding processor load, memory usage, available memory space, and other similar <u>information that is descriptive of the operation of the local workstation or server</u>. The development of such code is well known in the art, and any suitable code can be practiced with the invention without departing from the scope thereof. Also, each agent can

> <u>take action on the local system</u> such as setting off alarms, activating a beeper, sending a fax via modem, sending e-mail to system administrators or taking corrective action.

Regarding independent claims 4, 6, 15, and 18, Agarwal does not teach the claim element of "searching the [input/output packet or packet] to determine the identity of a process that created the [input/output packet or packet] and storing in a trace log an entry representing the [request or data]." Agarwal merely discloses the storage of stream information describing the type of connection. In relevant part, Col. 8, lines 27-66 state:

> In operation, for each connected stream, the module 80 creates a context data structure for storing information. The context is then linked, as shown, to the corresponding STREAMS queues, read and write. <u>The context stores information about the connected stream that describes the type of connection</u>. The module 80 also opens a context data structure 94 for communicating to the agent 50. The module 80 can store data about event details within this context 94, and the agent 50 can read the data out, and pass the information to the correlation processor 64. The tap table 82 stores a list of the connected streams context data structures, the list of all servers being monitored, and the context data structure for communicating with the agent 50.

Agarwal does not teach the claim 4, 6, 15, and 18 element cited by the Office action. Agarwal's capturing information regarding the type of connection is not equivalent to the claimed element described above.

Because none of the independent claims 1, 4, 6, 10, 12, 15, and 18 are anticipated by Agarwal, neither are any of the present application's dependant claims. Therefore, because "each and every element" of the present application's claims are not present in Agarwal, the current § 102(e) rejections are improper.

<u>Arguments for patentability under § 103(a)</u>

Another issue of patentability raised by the Office action and which the applicant must overcome is whether the applicant's invention is non-obvious over Agarwal in view of Khanna. The applicant submits that claims 7-9, 19 and 20 meet the requirements of patentability set out by 35 U.S.C. § 103(a) and are therefore allowable.

To establish a *prima facie* case of obviousness, and hence to find claims 7-9, 19 and 20 unpatentable under 35 U.S.C. § 103(a), three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all of the claim limitations. MPEP at § 2142 (emphasis added).

Combining Agarwal and Khanna does not teach or suggest all of the claim limitations of claim 7. First, while claim 7 describes creating a connection object representing the opening of the port connection by the process, Agarwal describes autonomous module agents which are "autopush[ed] on top of the TCP STREAMS stack so that all TCP traffic in and out of the managed node passes through the module." Col. 8, lines 11-14. Furthermore, other embodiments of Agarwal's agent structures autonomously "monitor all TCP traffic in and out of the managed node" through the "TAP driver" and are not created as a representation of the opening of the port connection by the process as described in claim 7. Col. 9, lines 16-17. Accordingly, the "creating a connection object representing the opening of the port connection by the process" element is not taught or suggested.

Likewise, the creation of a connection object which uses the well-known Transport Control Block (TCB) to create a log of network activity as claimed in claim 7 is not present in Agarwal or Khanna. In Khanna, the well-known TCB structure is described as merely keeping track of network connections. As claimed in claim 7, the created connection object first records each process identification into the TCB, then, once the connection object detects data at the port, the claimed object further copies the process identification into a trace log which is separate from the well-known TCB.

The TCB structure explained by Khanna is well-understood in the art as an essential element of the TCP package. All TCP devices use the TCB to control connections by keeping track of critical information including the process identification. However, because the TCB is only integral to the connection control functioning of TCP, it cannot alone be considered a proper structure for monitoring network performance in association with a created connection object as claimed. It is true that the TCB is a representation of information relating to TCP connections,

however, the TCB merely serves as a control structure for TCP's proper performance and not as a record of network activity.

Therefore, while Agarwal in view of Khanna's description of the TCB may motivate keeping track of network connections, it does not motivate monitoring the volume and flow of information across the network through the use of "a connection object" and "a trace log" as claimed in claim 7. Accordingly, the claim 7 elements of "creating <u>a connection object</u> representing the opening of the port connection by the process; copying the process identification from the connection object into a transport control block associated with the port; and in response to the detection of the receipt of data at the port, copying the process identification into the <u>trace log</u>" are not taught or suggested by the cited references. Because the proposed combination does not teach or suggest all of the claim 7 limitations, claims 8 and 9 dependent thereon do not contain these limitations, either.

Furthermore, independent claims 6 and 18 teach tracing each individual packet across the TCP stack rather than merely the existence of network connections as recorded in the TCB and described by Khanna. Claims 6 and 18 explicitly state that the method or instruction steps for tracing data do so for each packet sent across the network, not the mere existence of the network connection. Specifically, these claims describe tracing network performance by:

> ...detecting a transport-layer <u>request to transmit a packet</u> for an input/output connection to a port; <u>searching the packet</u> to determine an identity of a process that created <u>the packet</u>; and in response to the detection of a receipt of data at the port, <u>storing in a trace log an entry representing the receipt of the data</u>, wherein the entry comprises the process identification, and wherein the trace log is accessible <u>to determine a volume of the data</u> being transmitted over the network.

The TCB structure described in Khanna, which is well-known in the art, does not trace each packet as it crosses the TCP stack. Rather, as previously discussed, the TCB exists only as a control structure for each connection on the network. The information contained in the TCB traces each specific connection only and not the individual packet traffic as claimed in claims 6 and 18. Therefore, the above-quoted elements of both claims 6 and 18 are not present in the proposed combination.

The combination of Agarwal and Khanna does not teach or suggest all of the limitations found in independent claims 6 and 18. Therefore, none of the remaining dependant claims 7, 19, and 20 to which the Office action objects under § 103(a) contain these limitations. Also, because the proposed combination does not teach or suggest all of the claim 7 limitations, claims 8 and 9 dependent thereon are allowable as well.

Conclusion

In view of the above remarks, the applicant respectfully requests favorable reconsideration and passage to issuance of this application. The applicant invites the examiner to contact the undersigned attorney with any questions regarding this response or the application as a whole. If there are any additional fees or refunds required, the Commissioner is directed to charge or debit Deposit Account No. 13-2855.

Respectfully submitted for,
MARSHALL, GERSTEIN & BORUN LLP

November 3, 2005

By: WJ Kramer

William J. Kramer
Reg. No. 46,229
MARSHALL, GERSTEIN & BORUN LLP
233 South Wacker Drive
6300 Sears Tower
Chicago, Illinois 60606-6357
(312) 474-6300